Transportation Logistics

Part III: Traveling salesman problems

Motivation

Why do we study the TSP?

- it easy to formulate
- it is a difficult problem
- many significant real-world problems can be formulated as TSP

Application areas of the TSP

Vehicle routing

Once the assignment of customers to routes has been done. A TSP (possibly with side constraints) has to be solved for each vehicle.

Cutting wallpaper

Assume n sheets of wallpaper shall be cut from a single roll of paper. The amount of paper that is wasted varies depending on which sheet j is cut from the roll directly after i. We want to minimize the total wastage.

Application areas of the TSP

Job sequencing

n jobs shall be scheduled on a single machine. The jobs can be done in any order. Assume that we have sequence dependent setup times, i.e. depending on which job i precedes job j the time needed to adapt the machine to be able to process job j varies. All jobs shall be completed in the shortest possible time.

Clustering a data array

Computer wiring

...

The asymmetric TSP

$$x_{ij} = \begin{cases} 1, \text{ if arc } (ij) \text{ is part of the solution,} \\ 0, \text{ otherwise.} \end{cases}$$

$c_{ij} =$ the costs to traverse arc (i, j)

A...set of arcs,
V...set of vertices
(TSP is formulated on a complete directed graph)

IP Formulation

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \to \min \tag{1}$$

$$\sum_{i\in V\setminus\{j\}} x_{ij} = 1 \qquad \forall j \in V, \tag{2}$$

$$\sum_{j\in V\setminus\{i\}} x_{ij} = 1 \qquad \forall i \in V, \tag{3}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad \forall S \subset V, |S| \geq 2, \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A. \tag{5}$$

The asymmetric TSP - connectivity/subtour elimination

Option 1 (connectivity constraints)

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 \qquad \forall S \subset V, |S| \geq 2 \qquad (6)$$

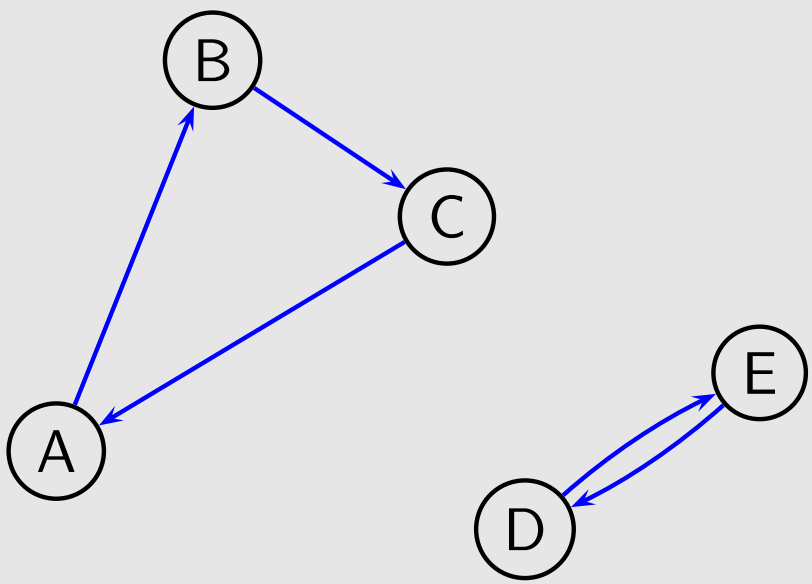
Option 2 (subtour elimination constraints)

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad \forall S \subset V, |S| \geq 2 \qquad (7)$$

These two formulations are algebraically equivalent. They prevent the formation of subtours containing fewer than $|S|$ vertices.
Note that the number of subtour elimination constraints needed is $2^{|V|} - |V| - 2$.

The asymmetric TSP

A solution with 2 subcycles



Subtour elim. for subtour A-B-C:
Option 1: $x_{AD} + x_{AE} + x_{BD} + x_{BE} + x_{CD} + x_{CE} \geq 1$
Option 2:
$x_{AB} + x_{AC} + x_{BA} + x_{BC} + x_{CA} + x_{CB} (+x_{AA} + x_{BB} + x_{CC}) \leq 2$

Subtour elim. for subtour D-E:
Option 1: $x_{DA} + x_{EA} + x_{DB} + x_{EB} + x_{DC} + x_{EC} \geq 1$
Option 2:
$x_{DE} + x_{ED} (+x_{DD} + x_{EE}) \leq 1$

A lower bound for the asymmetric TSP

The ATSP is known to be NP-hard (we do not know of a polynomial time algorithm for its solution).

A good **lower bound** on the optimal solution of the ATSP can be obtained by removing the subtour elimination constraints. The relaxed problem is the **Linear Assignment Problem!**

($c_{ii} := \infty$ such that $x^*_{ii} = 0$)

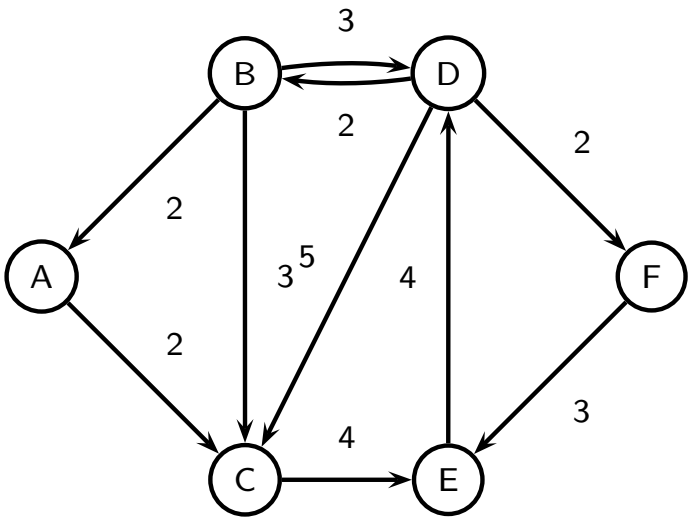
Rule: z^*_{AP} is a good lower bound on z^*_{ATSP} if the cost matrix is strongly asymmetric. (empirical tests showed that $(z^*_{ATSP} - z^*_{AP})/z^*_{AP}$ often $< 1\%$). If the cost matrix is symmetric, $(z^*_{ATSP} - z^*_{AP})/z^*_{AP}$ is often more than 30%.

Source: Laporte et al. (2004) Introduction to Logistics Systems Planning and Control, p. 252ff

ATSP: Patching heuristic

- **Initialization**
  Solve the AP. Let $C = \{C_1, \ldots, C_p\}$ denote the set of subcycles in the optimal solution of the AP. If $|C| = 1 \rightarrow$ STOP. Otherwise proceed with step 2.
- **Step 1**
  Identify the two subcycles with the largest number of vertices.
- **Step 2**
  Merge these two subcycles such that the cost increase is as small as possible and update C. If $|C| = 1 \rightarrow$ STOP. Otherwise go back to step 2.

ATSP: Patching heuristic - Example



c_{ij}	A	B	C	D	E	F
A	0	12	2	10	6	12
B	2	0	3	3	7	5
C	12	10	0	8	4	10
D	4	2	5	0	5	2
E	8	6	9	4	0	6
F	11	9	12	7	3	0

Initialization apply the Hungarian method. Several alternative AP solutions with Z = 23:

Solution 1:

$A \to C \to F \to E \to D \to B \to A$

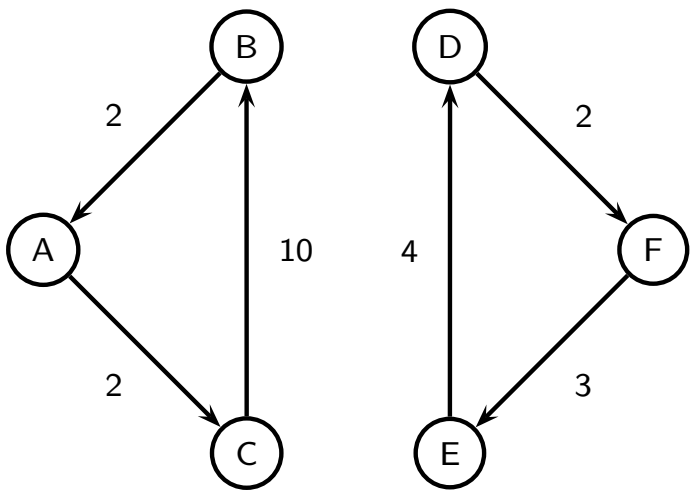
Solution 2:

$A \to C \to E \to B \to A$, $D \to F \to D$

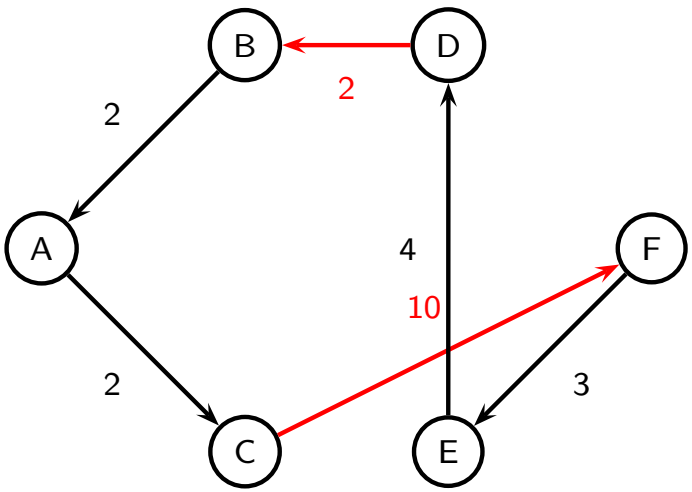
Solution 3:

$A \to C \to B \to A$, $D \to F \to E \to D$

...



ATSP: Patching heuristic - Example



c_{ij}	A	B	C	D	E	F
A	0	12	2	10	6	12
B	2	0	3	3	7	5
C	12	10	0	8	4	10
D	4	2	5	0	5	2
E	8	6	9	4	0	6
F	11	9	12	7	3	0

Steps 1 and 2 Find the best way to join the two subcycles:

Insert 2nd tour between A and C:

$A \to D \to F \to E \to C \quad \Delta = 10 + 9 - 2 - 4 = 13$

$A \to F \to E \to D \to C \quad \Delta = 12 + 5 - 2 - 2 = 13$

$A \to E \to D \to F \to C \quad \Delta = 6 + 12 - 2 - 3 = 13$

Insert 2nd tour between C and B:

$C \to D \to F \to E \to B \quad \Delta = 8 + 6 - 10 - 4 = 0$

$C \to F \to E \to D \to B \quad \Delta = 10 + 2 - 10 - 2 = 0$

$C \to E \to D \to F \to B \quad \Delta = 4 + 9 - 10 - 3 = 0$

Insert 2nd tour between B and A:

$B \to D \to F \to E \to A \quad \Delta = 3 + 8 - 2 - 4 = 5$

$B \to F \to E \to D \to A \quad \Delta = 5 + 4 - 2 - 2 = 5$

$B \to E \to D \to F \to A \quad \Delta = 7 + 11 - 2 - 3 = 13$

e.g. best solution (several):

$A \to C \to F \to E \to D \to B \to A$

$Z = 23$ (upper bound on z^*_{ATSP})

The symmetric TSP (STSP)

Since lower and upper bounding procedures for the ATSP do not perfom well if applied to the STSP, procedures tailored to the STSP have been developed.

Notation

$$x_e = \begin{cases} 1, \text{ if edge } e \text{ is part of the solution,} \\ 0, \text{ otherwise.} \end{cases}$$

$$c_e = c_{ij} = c_{ji}$$

E...edge set

STSP: IP formulations

$$\sum_{(i,j)\in E} x_{ij}c_{ij} \to \min \tag{8}$$

$$\sum_{j\in V:(j,i)\in E} x_{ji} + \sum_{j\in V:(i,j)\in E} x_{ij} = 2 \quad \forall i \in V, \tag{9}$$

$$\sum_{(i,j)\in E:i\in S,j\notin S} x_{ij} + \sum_{(j,i)\in E:i\in S,j\notin S} x_{ji} \geq 2 \quad \forall S \subset V, 2 \leq |S| \leq \lceil |V|/2 \rceil, \tag{10}$$

$$x_e \in \{0,1\} \qquad \forall e \in E. \tag{11}$$

$i < j$ for each edge $(i,j) \in E$

Since the connectivity constraints of subset S are equivalent to those of subset $V \setminus S$, we only consider $S \subset V$ such that $|S| \leq \lceil |V|/2 \rceil$.

STSP: IP formulations

$$\sum_{e \in E} x_e c_e \to \min \tag{12}$$

$$\sum_{e \in J(i)} x_e = 2 \qquad \forall i \in V, \tag{13}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \qquad \forall S \subset V, 2 \leq |S| \leq \lceil |V|/2 \rceil \tag{14}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E. \tag{15}$$

$J(i)$ all edges connected to i; $E(S)$ edges connecting vertices in subset S

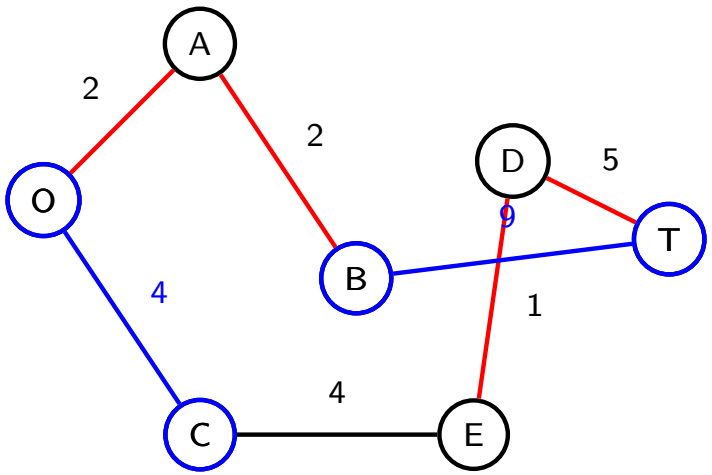
STSP: a lower bound

The STSP is also NP-hard. A valid **lower bound** on the optimal solution cost z^*_{STSP} is **the optimal solution cost of the MST** z^*_{MST}

STSP: Christofides' heuristic

- **Initialization** Compute a minimum spanning tree T.
- **Step 1** Compute a least-cost perfect matching among the vertices of odd degree in T. (V_D...vertices with odd degree). Add the edges of the perfect matching (M...vertices of the matching, H...graph induced by union of the edges $\in T$ and $\in M$).
- **Step 2** If there is a vertex $j \in V$ of degree > 2, eliminate two edges incident in j, denote them (j,k),(j,h) with $k \neq h$ and add edge (k,h) (try to replace those two edges that improve the solution value the most and do not lead to subcycles). Repeat step 2 until all vertices in V have a degree of 2.

STSP: Christofides' heuristic - Example



	O	A	B	C	D	E	T
O	0	2	4	4	8	7	13
A	2	0	2	3	6	5	11
B	4	2	0	1	4	3	9
C	4	3	1	0	5	4	10
D	8	6	4	5	0	1	5
E	7	5	3	4	1	0	6
T	13	11	9	10	5	6	0

Initialization compute the minimum spanning tree (Kruskal's algorithm!)

Step 1 $V_D = \{O, C, B, T\}$
Compute a least-cost perfect matching:
$O - B, C - T$: $4 + 10 = 14$
$O - C, B - T$: $4 + 9 = 13$
$O - T, B - C$: $13 + 1 = 14$
Least cost perfect matching:
$E(M) = \{(O, C), (B, T)\}$: $4 + 9 = 13$

Step 2 $j = B$ find shortcuts.
B is connected to A,C,E,T
possible shortcuts:
$A - E$: $\Delta = 5 - 5 = 0$
$A - T$: $\Delta = 11 - 11 = 0$
$C - E$: $\Delta = 4 - 4 = 0$
$C - T$: $\Delta = 10 - 10 = 0$
We choose $C - E$
$z_{STSP} = 27$
$z^*_{MST} = 14$

Solution methods

Heuristic methods

- Nearest neighbor
- Cheapest insertion
- k-opt (local search)

Exact methods

- Branch and bound algorithms

Nearest neighbor heuristic

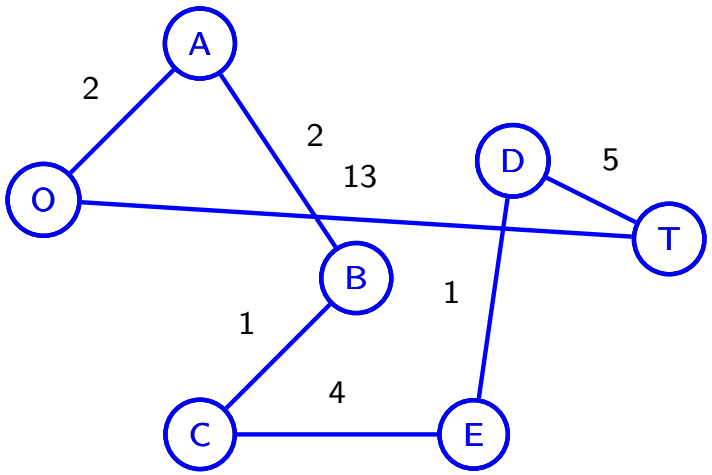
- **Initialization** Set $P = \{r\}$ with r being a vertex chosen arbitrarily and P denotes the (partial) TSP tour. Set $h = r$.
- **Step 1** Find the vertex $k \in V \setminus P$ such that $c_{hk} = \min_{j \in V \setminus P}\{c_{hj}\}$. Append k to the end of P.
- **Step 2** If $|P| = |V|$, add r to the end of P and STOP. (P is now a Hamiltonian cycle). Otherwise, set $h = k$ and continue with step 1.

Hamiltonian cycle

A cycle that contains all the vertices of a graph. (TSP = problem of finding the minimum cost Hamiltonian cycle in a graph)

Sources: Laporte et al. (2004) Introduction to Logistics Systems Planning and Control, p. 261
E.L. Lawler J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds) (1990) 'The traveling salesman problem', p. 2, p. 150f.

Nearest neighbor heuristic - Example



	O	A	B	C	D	E	T
O	0	2	4	4	8	7	13
A	2	0	2	3	6	5	11
B	4	2	0	1	4	3	9
C	4	3	1	0	5	4	10
D	8	6	4	5	0	1	5
E	7	5	3	4	1	0	6
T	13	11	9	10	5	6	0

Initialization $r = O$, $P = \{O\}$, $h = O$

Step 1 identify the nearest neighbor to O
$\rightarrow A$: $k = A$, $P = \{O, A\}$, $Z = 2$
Step 2 $|P| < |V| \rightarrow$
Step 1 identify the nearest neighbor to A
$\rightarrow B$: $k = B$, $P = \{O, A, B\}$, $Z = 4$
Step 2 $|P| < |V| \rightarrow$
Step 1 identify the nearest neighbor to B
$\rightarrow C$: $k = C$, $P = \{O, A, B, C\}$, $Z = 5$
Step 2 $|P| < |V| \rightarrow$
...
$P = \{O, A, B, C, E, D, T\}$,
$Z = 5 + 4 + 1 + 5 = 15$
Step 2 $|P| = |V|$, append O to P
$P = \{O, A, B, C, E, D, T, O\}$
$z_{TSP} = 28$

Nearest neighbor heuristic - Observations

Advantages

Simple and fast

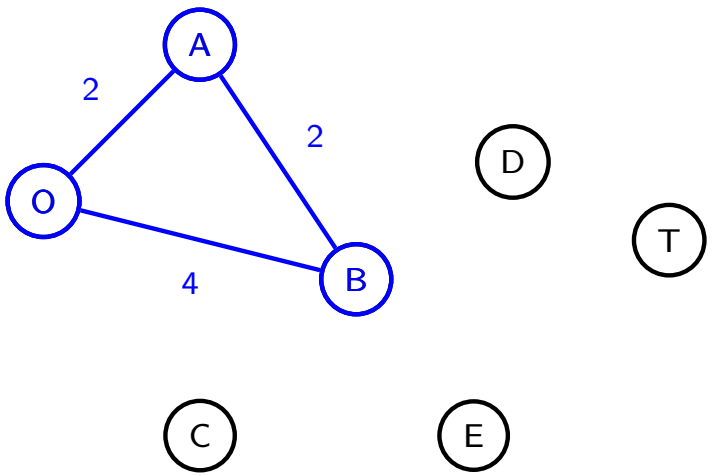
Disadvantages

Solution can be of rather poor quality. (edges added in later iterations may be quite long - especially the last edge connecting back to the first vertex)

Cheapest insertion heuristic

- **Initialization** arbitrarily choose the first vertex r. Initialize the TSP tour $P = \{r, r\}$, $Z = 0$
- **Iteration step** If $|P| < |V| + 1$
  - For each vertex $i \in V \setminus P$ find the cheapest insertion position between any j and k; $j, k \in P$ and neighboring in P.
  - Find the vertex $i^* \in V \setminus P$ that can be inserted the cheapest; insert it at its cheapest position (say j^* and k^*):
    $P = \{r, \ldots, j^*, i^*, k^*, \ldots, r\}$ and set
    $Z = Z - c_{j^*k^*} + c_{j^*i^*} + c_{i^*k^*}$

Cheapest insertion heuristic - Example



	O	A	B	C	D	E	T
O	0	2	4	4	8	7	13
A	2	0	2	3	6	5	11
B	4	2	0	1	4	3	9
C	4	3	1	0	5	4	10
D	8	6	4	5	0	1	5
E	7	5	3	4	1	0	6
T	13	11	9	10	5	6	0

Initialization $r = O,\ P = \{O, O\}\ Z = 0$

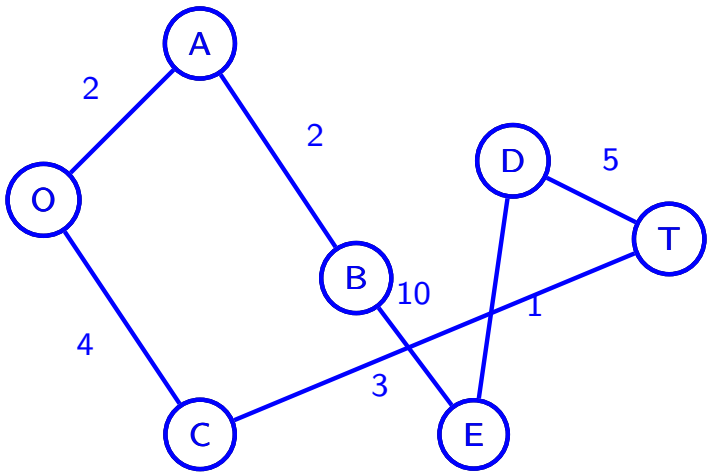
Iteration 1

$O - A - O\ \Delta = 2 + 2 = 4$
$O - B - O\ \Delta = 4 + 4 = 8$
$O - C - O\ \Delta = 4 + 4 = 8$
$O - D - O\ \Delta = 8 + 8 = 16$
$O - E - O\ \Delta = 7 + 7 = 14$
$O - T - O\ \Delta = 13 + 13 = 26$
A: $i^* = A,\ P = \{O, A, O\},\ Z = 4$

Iteration 2

$O - B - A,\ A - B - O$
$\Delta = -2 + 2 + 4 = 4$
$O - C - A,\ A - C - O$
$\Delta = -2 + 4 + 3 = 5$
$O - D - A,\ A - D - O$
$\Delta = -2 + 8 + 6 = 12$
$O - E - A,\ A - E - O$
$\Delta = -2 + 7 + 5 = 10$
$O - T - A,\ A - T - O$
$\Delta = -2 + 13 + 11 = 24$
B: $i^* = B,\ P = \{O, B, A, O\},\ Z = 8$

Cheapest insertion heuristic - Example



	O	A	B	C	D	E	T
O	0	2	4	4	8	7	13
A	2	0	2	3	6	5	11
B	4	2	0	1	4	3	9
C	4	3	1	0	5	4	10
D	8	6	4	5	0	1	5
E	7	5	3	4	1	0	6
T	13	11	9	10	5	6	0

Iteration 3

$O - C - B\ \Delta = -4 + 4 + 1 = 1$
$B - C - A\ \Delta = -2 + 1 + 3 = 2$
$A - C - O\ \Delta = -2 + 4 + 3 = 5$
$O - D - B\ \Delta = -4 + 8 + 4 = 8$
$B - D - A\ \Delta = -2 + 4 + 6 = 10$
$A - D - O\ \Delta = -2 + 8 + 6 = 12$

...

C: $i^* = C$, $P = \{O, C, B, A, O\}$, $Z = 9$

...

$P = \{O, C, T, D, E, B, A, O\}$, $z_{TSP} = 27$

Local search

Local search

Local search algorithms are iterative procedures that try to improve an initial feasible solution $x^{(0)}$. At the kth iteration, all solutions contained in a 'neighborhood' of the current solution $x^{(k)}$ are enumerated. If there exist feasible solutions that are less costly than $x^{(k)}$, the best solution of the neighborhood becomes the new current solution $x^{(k+1)}$ and the procedure is repeated. Otherwise, the procedure ends. The last current solution is the local optimum.

Source: Laporte et al. (2004) Introduction to Logistics Systems Planning and Control, p. 263

Neighborhood

A neighborhood is defined by all feasible solutions that can be reached by applying a given operator to a given solution (e.g.: move one vertex, exchange two vertices).

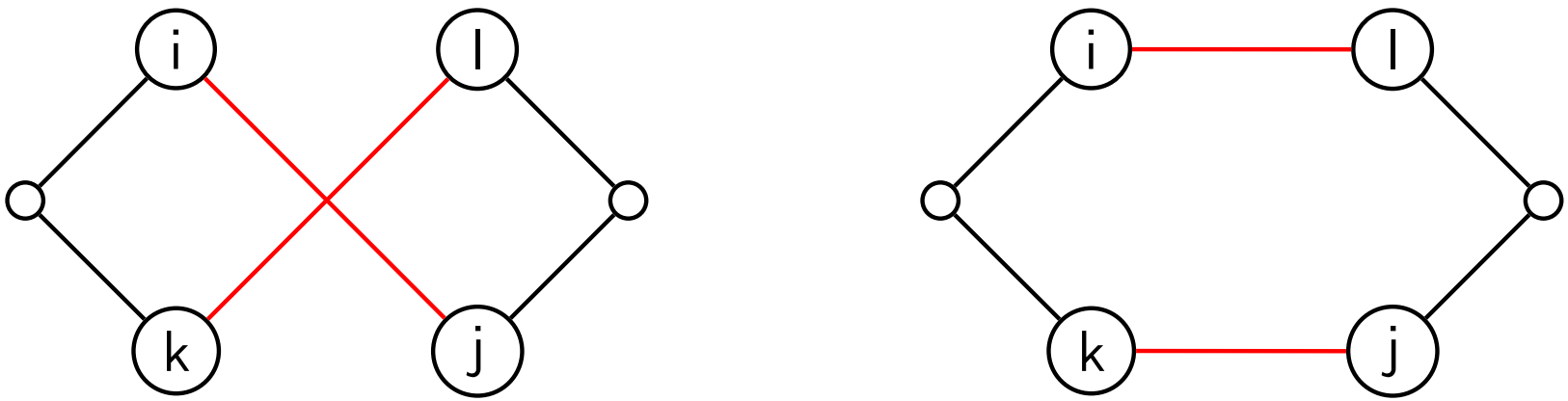
Lin-Kernighan: k-opt (local search)

- **Initialization** Let $P^{(0)}$ be the initial TSP solution and let $z_{TSP}^{(0)}$ be the cost of $P^{(0)}$. Set $h = 0$.
- **Step 1** Identify the best feasible solution $P^{(h+1)}$ that can be obtained through a k-exchange. If $z_{TSP}^{(h+1)} \geq z_{TSP}^{(h)}$, STOP.
- **Step 2** set $h = h + 1$ and proceed with step 1.

k-exchange

replace k links (edges) of the current solution with k new links, e.g. a 2-exchange: replace (i, j) and (k, l) by (i, l) and (j, k) (= inverse the sequence between j and k)

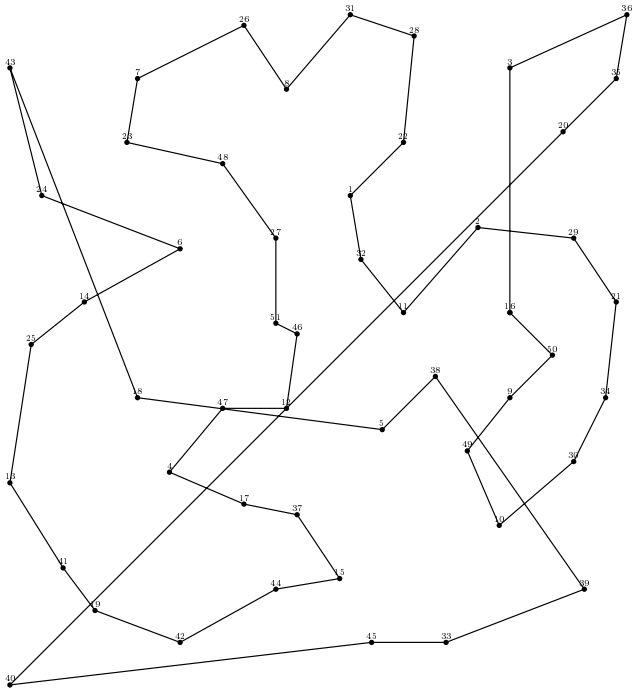
Lin-Kernighan: 2-exchange



Lin-Kernighan: 2-opt (typical picture)

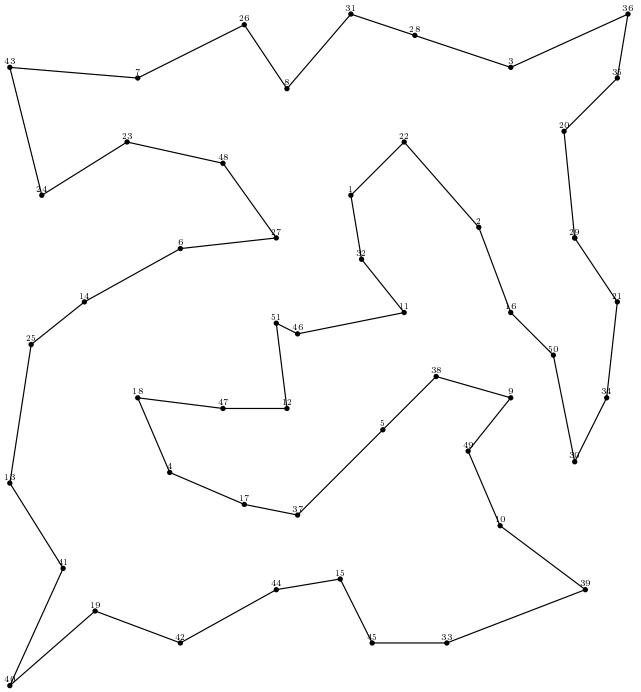
16 - 50 - 9 - 49 - 10 - 30 - 34 - 21 - 29 - 2 - 11 - 32 - 1 - 22 - 28 - 31 - 8 - 26 - 7 - 23 - 48 - 27 - 51 - 46 - 12 - 47 - 4 - 17 - 37 - 15 - 44 - 42 - 19 - 41 - 13 - 25 - 14 - 6 - 24 - 43 - 18 - 5 - 38 - 39 - 33 - 45 - 40 - 20 - 35 - 36 - 3 - 16

Tourlength = 576.068

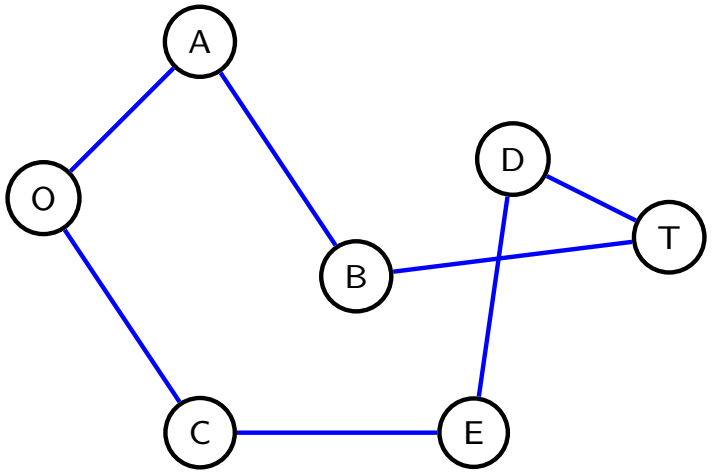


29 - 21 - 34 - 30 - 50 - 16 - 2 - 22 - 1 - 32 - 11 - 46 - 51 - 12 - 47 - 18 - 4 - 17 - 37 - 5 - 38 - 9 - 49 - 10 - 39 - 33 - 45 - 15 - 44 - 42 - 19 - 40 - 41 - 13 - 25 - 14 - 6 - 27 - 48 - 23 - 24 - 43 - 7 - 26 - 8 - 31 - 28 - 3 - 36 - 35 - 20 - 29

Tourlength = 436.123



Lin-Kernighan: 2-opt - Example



Initial solution

$O - C - E - D - T - B - A - O$

best improvement: the best one becomes the new current solution; repeat until no further improvement.

first improvement: the first improving solution becomes the new current solution; repeat until no further improvement.

2-opt

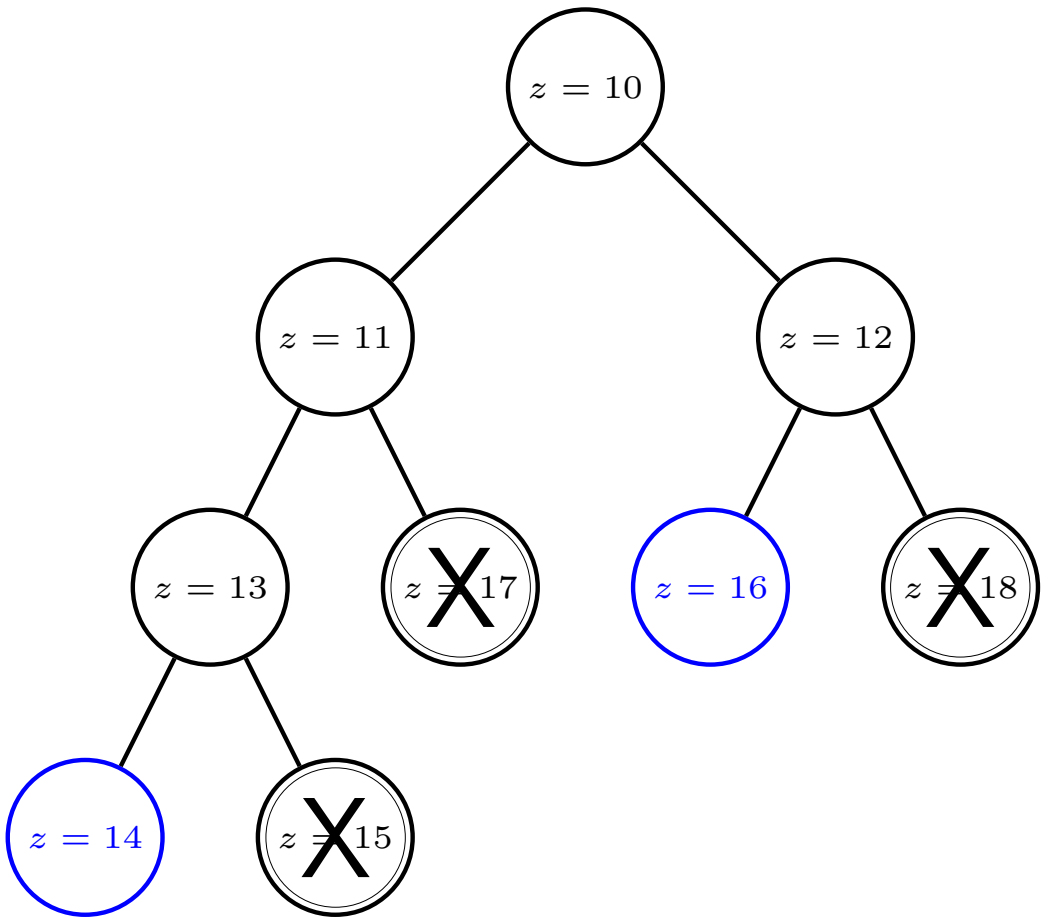
$O - E - C - D - T - B - A - O$
$O - C - D - E - T - B - A - O$
$O - C - E - T - D - B - A - O$
$O - C - E - D - B - T - A - O$
$O - C - E - D - T - A - B - O$
$O - D - E - C - T - B - A - O$
$O - C - T - D - E - B - A - O$
$O - C - E - B - T - D - A - O$
$O - C - E - D - A - B - T - O$
$O - T - D - E - C - B - A - O$
$O - C - B - T - D - E - A - O$
$O - C - E - A - B - T - D - O$
$O - B - T - D - E - C - A - O$
$O - C - A - B - T - D - E - O$
$O - A - B - T - D - E - C - O$

Branch and bound (BB) - the concept

A general method for solving (mixed) integer optimization problems (with a minimization objective). It uses two concepts: **branching** and **bounding**.

- **Root node** solve relaxation of the optimization problem (e.g., LP of IP). Set LB to the objective function value of the relaxed problem z_0 and set $UB = \infty$. Put the root node into the set of unprocessed nodes Q.
- **Repeat** (until Q is empty)
  - select a node k from the set Q, according to a certain criterion.
  - identify on what to **branch**
  - **branch**: partition the problem P_k at k into a given number of subproblems s: $P_{k_1} \dots P_{k_s}$ (one child node for each subproblem); for the solution spaces wrt the original optimization problem $L(P_k) = \cup_{i=1}^{s} P_{k_i}$ has to hold. (e.g., 2 child nodes: in the 1st child node (1st partition) a certain characteristic is forbidden while in the 2nd child node (2nd partition), this characteristic is enforced).
  - compute the lower bounds at the child nodes (new enforced/forbidden characteristics plus all enforced/forbidden characteristics of parent nodes); put the child nodes into Q
  - **bound:** the lowest bound across all nodes $\in Q$ gives the current LB; a feasible solution to the original problem at a given node gives a new UB; all open nodes associated with LBs $\geq UB$ can be pruned (removed from Q).

Branch and bound (BB) - the concept



$LB = 1011121314$
$UB = \infty 1614$

ATSP: Little et al. BB algorithm

θ_{ij} is the sum of the smallest cost element in row i (excluding c_{ij}) and the smallest cost element in column j (excluding c_{ij}); only computed for arcs with $c_{ij} = 0$.

arc that would lead to a cycle: e.g., already committed arcs (1,2),(3,4), new committed arc (2,3); arc that would lead to a cycle: (4,1);
e.g., already committed arcs (1,2),(3,4), new committed arc (5,6); arc that would lead to a cycle: (6,5).

ATSP: Little et al. BB algorithm

Lower bounds

let $z(t)$ denote the costs of a tour under a matrix before row- and column-wise reduction (tour $\rightarrow$ one entry in each row and column selected!); let $z_1(t)$ denote the cost under the matrix afterward and h the reduction constant:

$$z(t) = h + z_1(t)$$

therefore, h is a valid lower bound for $z(t)$.

If c_{ij} is set to M (forbidden), row i and column j can be reduced by their smallest entries; the sum of these two is given by θ_{ij}.

ATSP: Little et al. BB - Example

Root node $k = 0$

	1	**2**	**3**	**4**	**5**	**6**	
1	M	7	4	2	1	3	-1
2	3	M	3	2	4	6	-2
3	2	3	M	4	5	3	-2
4	7	1	5	M	4	4	-1
5	4	4	3	5	M	3	-3
6	4	3	3	6	2	M	-2

ATSP: Little et al. BB - Example

Root node $k = 0$

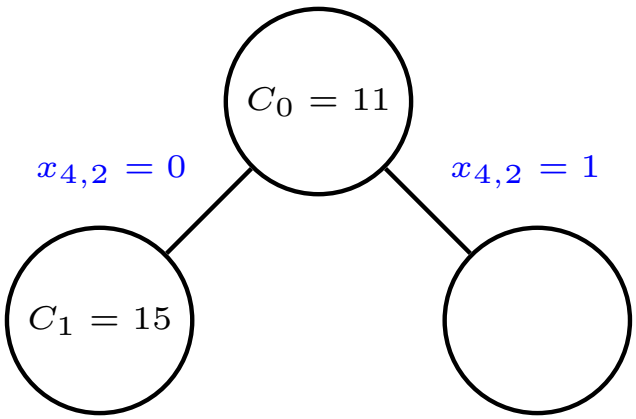
	1	**2**	**3**	**4**	**5**	**6**
1	M	6	3	1	0 (1)	2
2	1	M	1	0 (2)	2	4
3	0 (2)	1	M	2	3	1
4	6	0 (4)	4	M	3	3
5	1	1	0 (1)	2	M	0(1)
6	2	1	1	4	0 (1)	M

reduction constant = 1+2+2+1+3+2=11

$C_0 = 11$

branch on $(4, 2)$

ATSP: Little et al. BB - Example



ATSP: Little et al. BB - Example

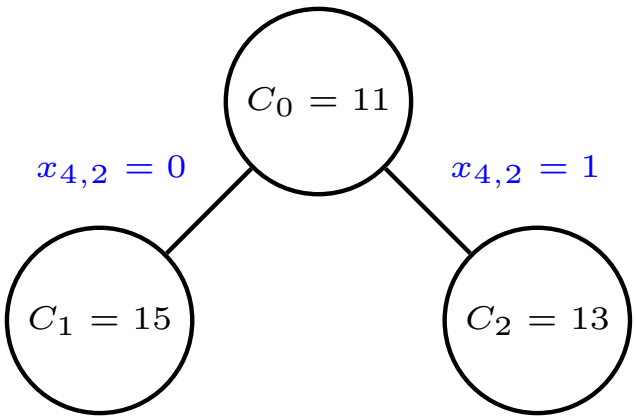
Compute bound at node 2

(4,2) has to be included (eliminate according row and column and set $c_{2,4} = M$)

	1	**3**	**4**	**5**	**6**	
1	M	3	1	0	2	
2	1	1	M	2	4	-1
3	0	M	2	3	1	
5	1	0	2	M	0	
6	2	1	4	0	M	
			-1			

$C_2 = C_0 + 2$

ATSP: Little et al. BB - Example



select node 2

ATSP: Little et al. BB - Example

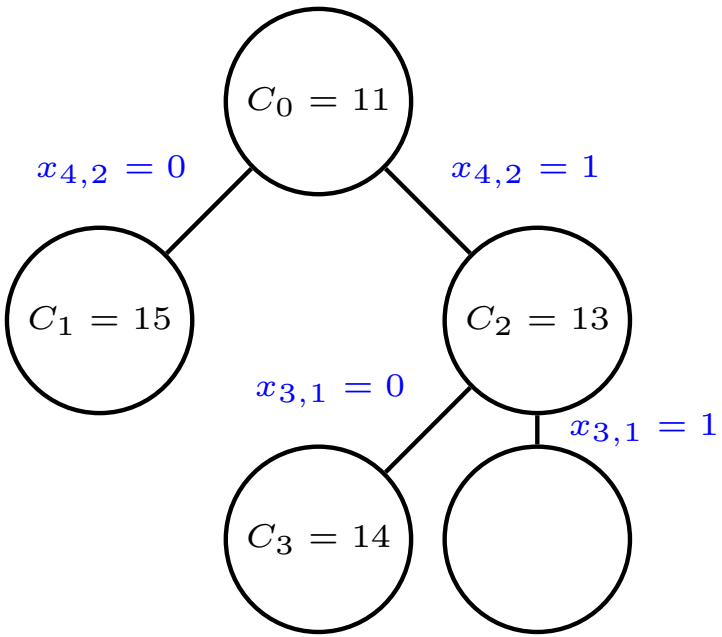
Decide on what to branch at node 2

compute the θ values.

	1	**3**	**4**	**5**	**6**
1	M	3	0(1)	0(0)	2
2	0(0)	0(0)	M	1	3
3	0(1)	M	1	3	1
5	1	0(0)	1	M	0(1)
6	2	1	3	0(1)	M

branch on (3,1)

ATSP: Little et al. BB - Example



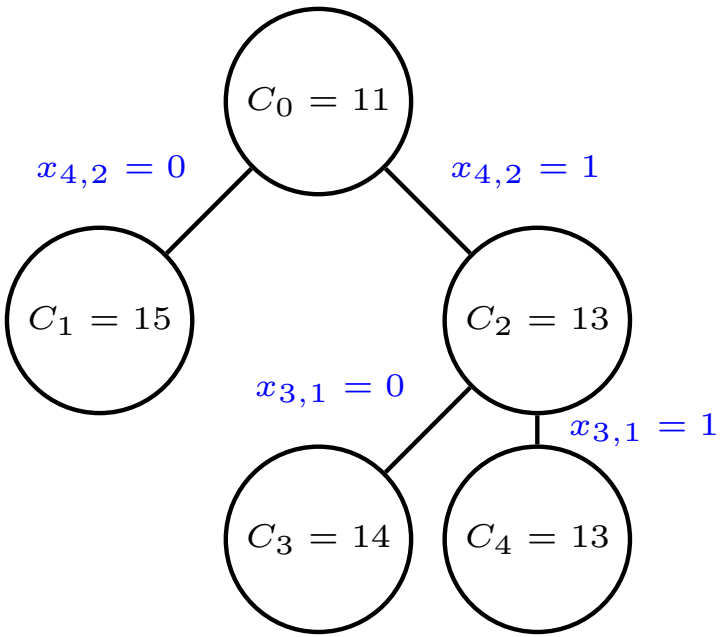
ATSP: Little et al. BB - Example

Compute the lower bound at node 4
include (3,1) and forbid (1,3)

	3	**4**	**5**	**6**
1	M	0	0	2
2	0	M	1	3
5	0	1	M	0
6	1	3	0	M

no reduction possible: $C_4 = C_2 = 13$

ATSP: Little et al. BB - Example



select node 4

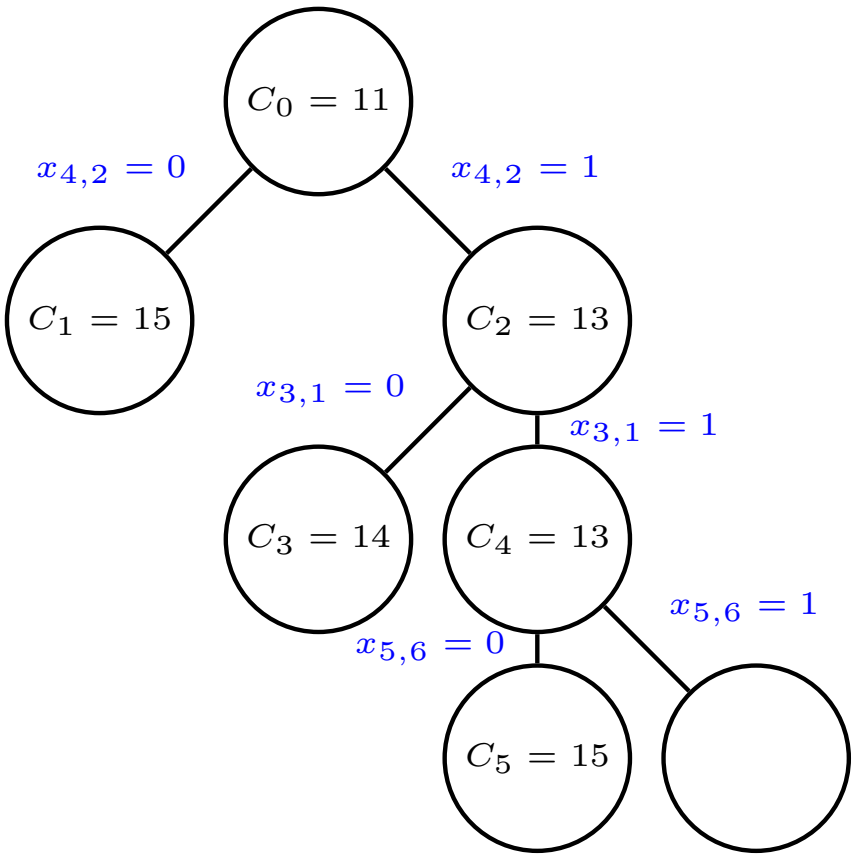
ATSP: Little et al. BB - Example

Decide on what to branch at node 4

	3	**4**	**5**	**6**
1	M	0 (1)	0 (0)	2
2	0 (1)	M	1	3
5	0 (0)	1	M	0 (2)
6	1	3	0 (1)	M

branch on (5,6)

ATSP: Little et al. BB - Example



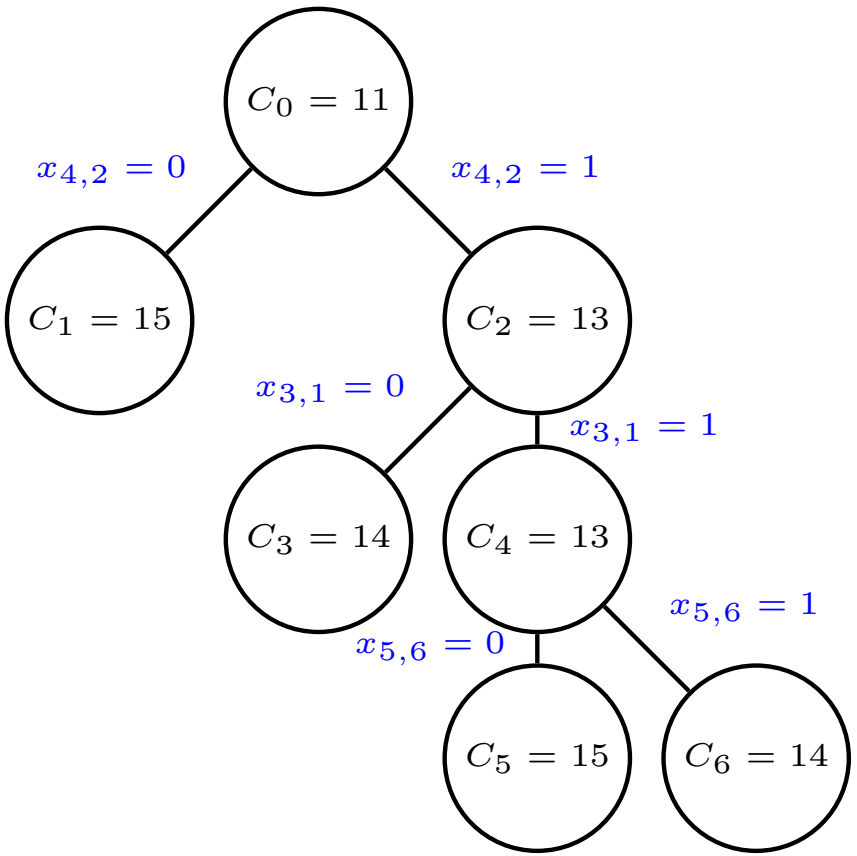
ATSP: Little et al. BB - Example

Compute the lower bound at node 6
include (5,6) and forbid (6,5)

	3	**4**	**5**	
1	M	0	0	
2	0	M	1	
6	1	3	M	-1

$C_6 = C_4 + 1 = 14$

ATSP: Little et al. BB - Example



select node 6

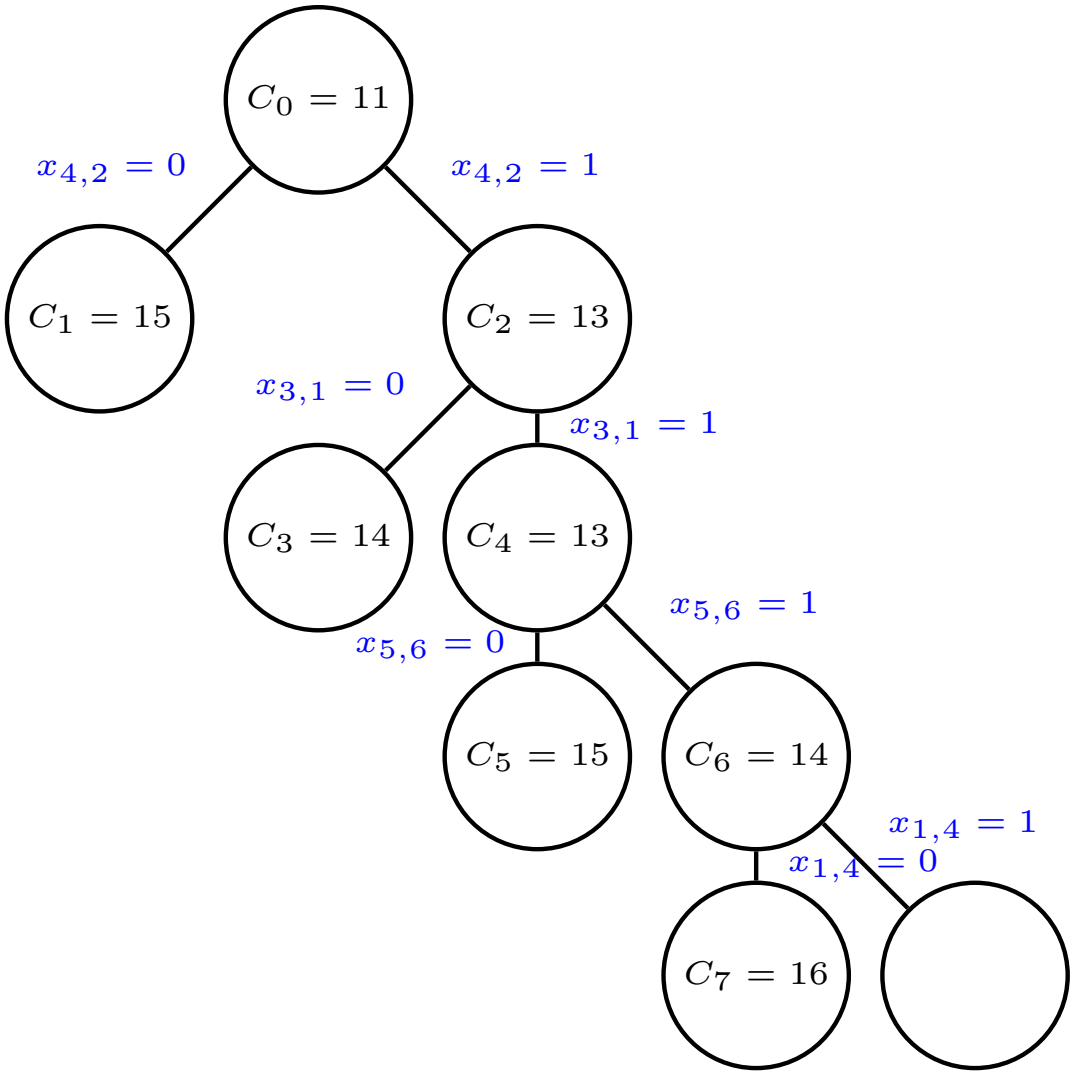
ATSP: Little et al. BB - Example

Decide on what to branch at node 6

	3	**4**	**5**
1	M	0 (2)	0 (1)
2	0 (1)	M	1
6	0 (2)	2	M

branch on (1,4)

ATSP: Little et al. BB - Example



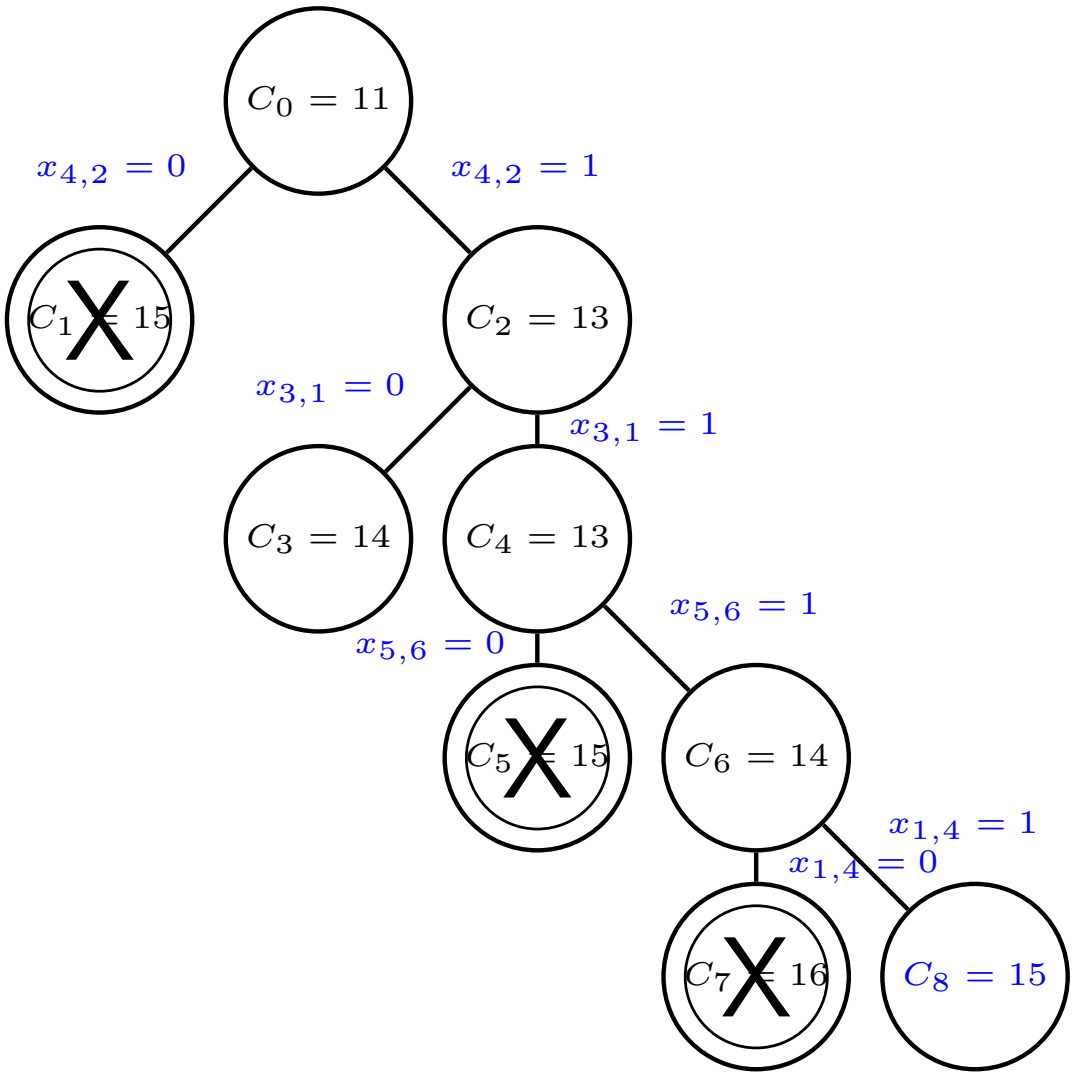
ATSP: Little et al. BB - Example

Compute the lower bound at node 8
include (1,4) and forbid (2,3)

	3	**5**
2	M	1
6	0	M
		-1

2x2 matrix: only two feasible arcs left to complete the tour!
$C_8 = C_6 + 1 = 15$
4-2-5-6-3-1-4

ATSP: Little et al. BB - Example



TSP solution at node 8 $\rightarrow$ prune nodes 1, 5 and 7 and select node 3

ATSP: Little et al. BB - Example

Decide on what to branch at node 3

$(x_{42} = 1,\ x_{3,1} = 0)$

	1	**3**	**4**	**5**	**6**	
1	M	3	0	0	2	
2	0	0	M	1	3	
3	M	M	1	3	1	-1
5	1	0	1	M	0	
6	2	1	3	0	M	

ATSP: Little et al. BB - Example

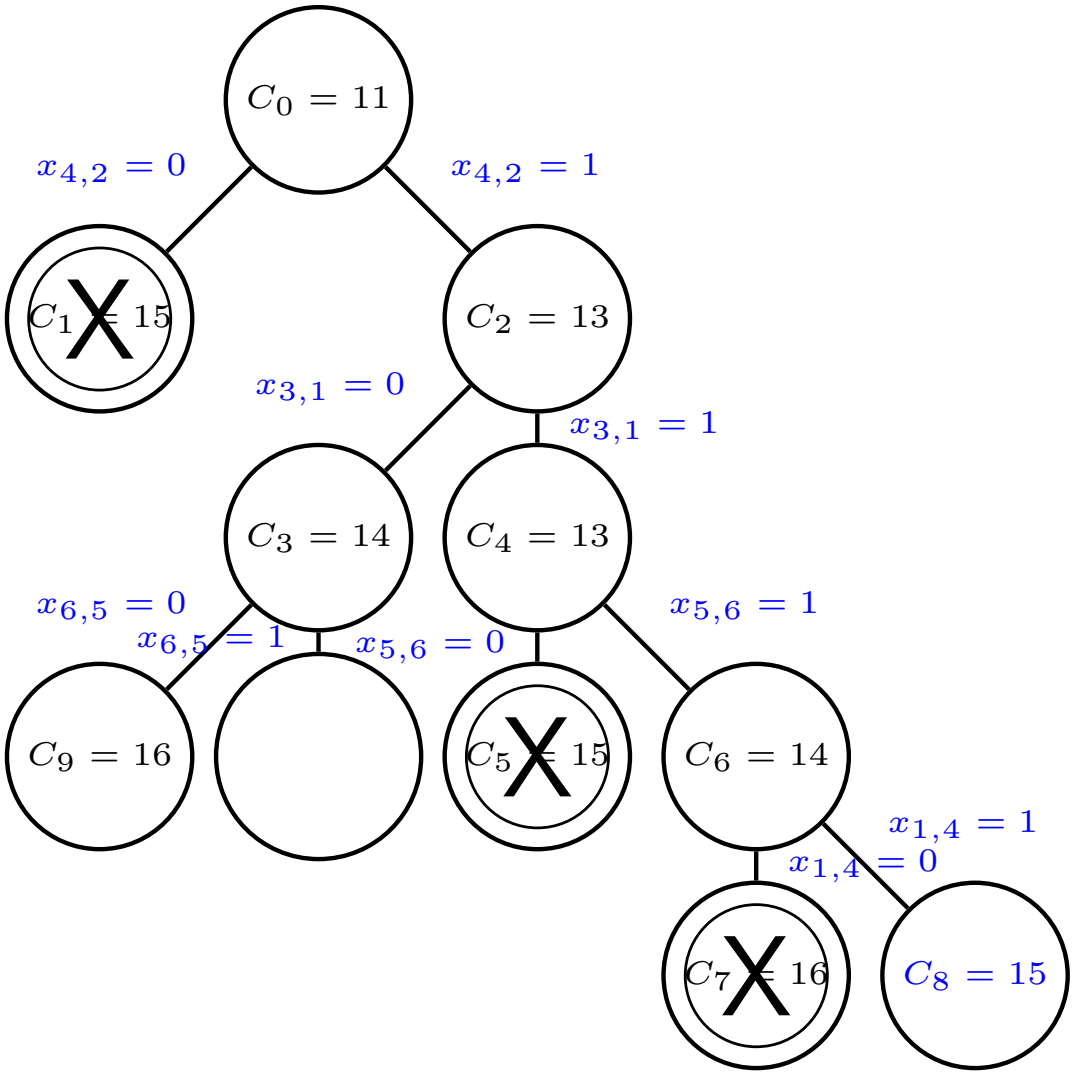
Decide on what to branch at node 3

$(x_{4,2} = 1,\ x_{3,1} = 0)$

	1	**3**	**4**	**5**	**6**
1	M	3	0 (0)	0 (0)	2
2	0 (1)	0 (0)	M	1	3
3	M	M	0 (0)	2	0 (0)
5	1	0 (0)	1	M	0 (0)
6	2	1	3	0 (1)	M

branch on (6,5)

ATSP: Little et al. BB - Example



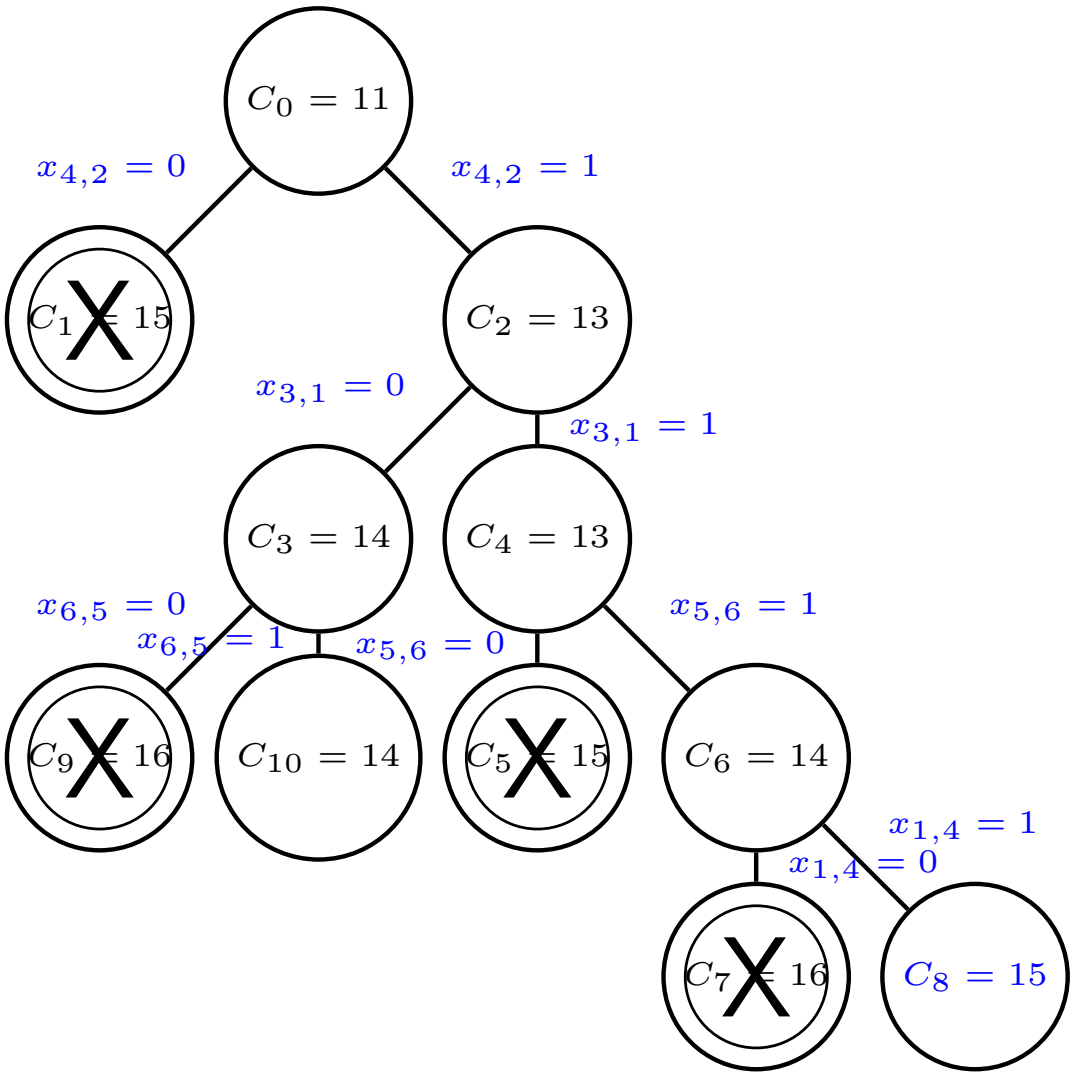
ATSP: Little et al. BB - Example

Compute the lower bound at node 10
(remove row 6 and column 5, set (5,6) to M)

	1	**3**	**4**	**6**
1	M	3	0	2
2	0	0	M	3
3	M	M	0	0
5	1	0	1	M

$C_{10} = C_3 = 14$

ATSP: Little et al. BB - Example



prune node 9 and select node 10

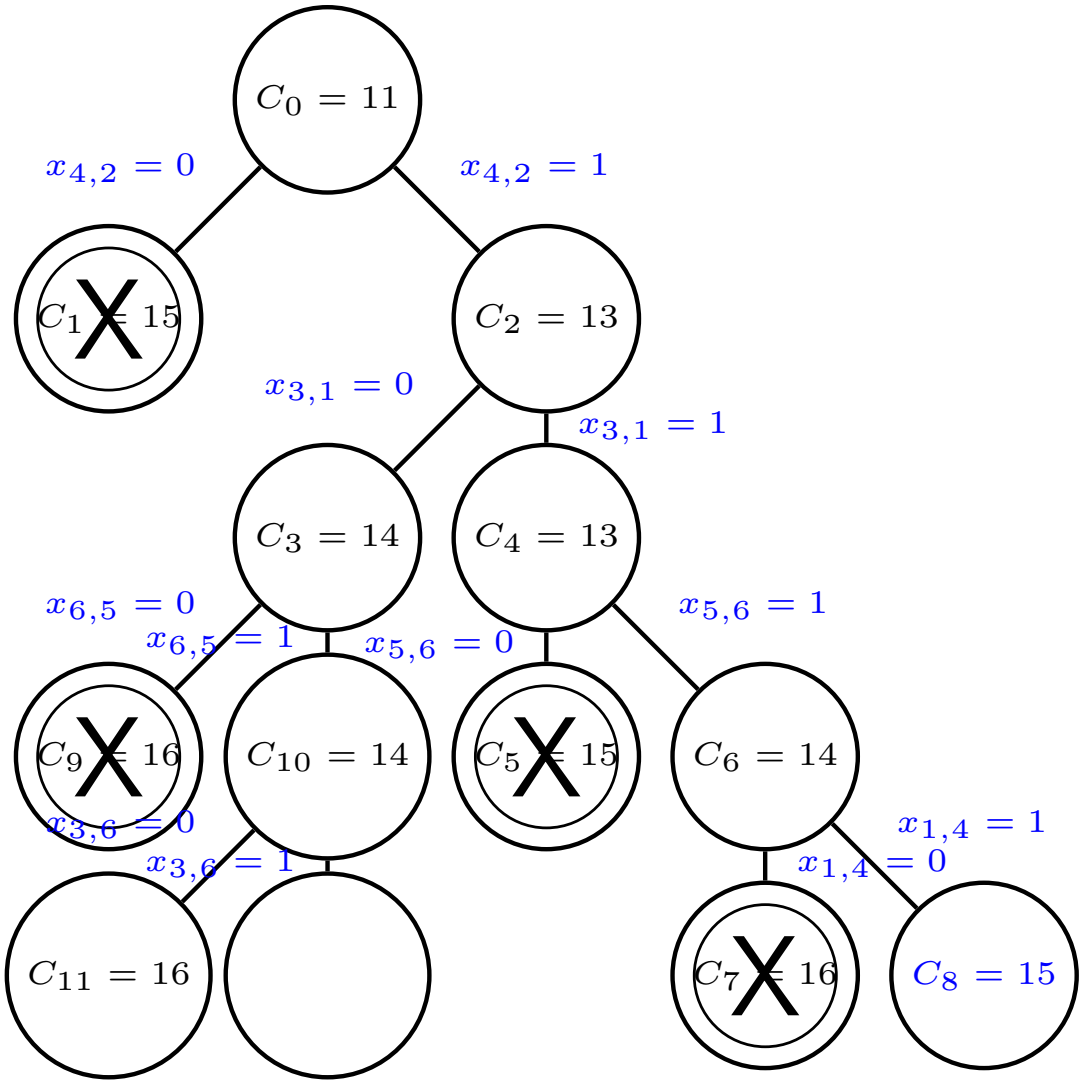
ATSP: Little et al. BB - Example

Decide on what to branch at node 10

	1	**3**	**4**	**6**
1	M	3	0 (2)	2
2	0 (1)	0 (0)	M	3
3	M	M	0 (0)	0 (2)
5	1	0 (1)	1	M

branch on (3,6)

ATSP: Little et al. BB - Example



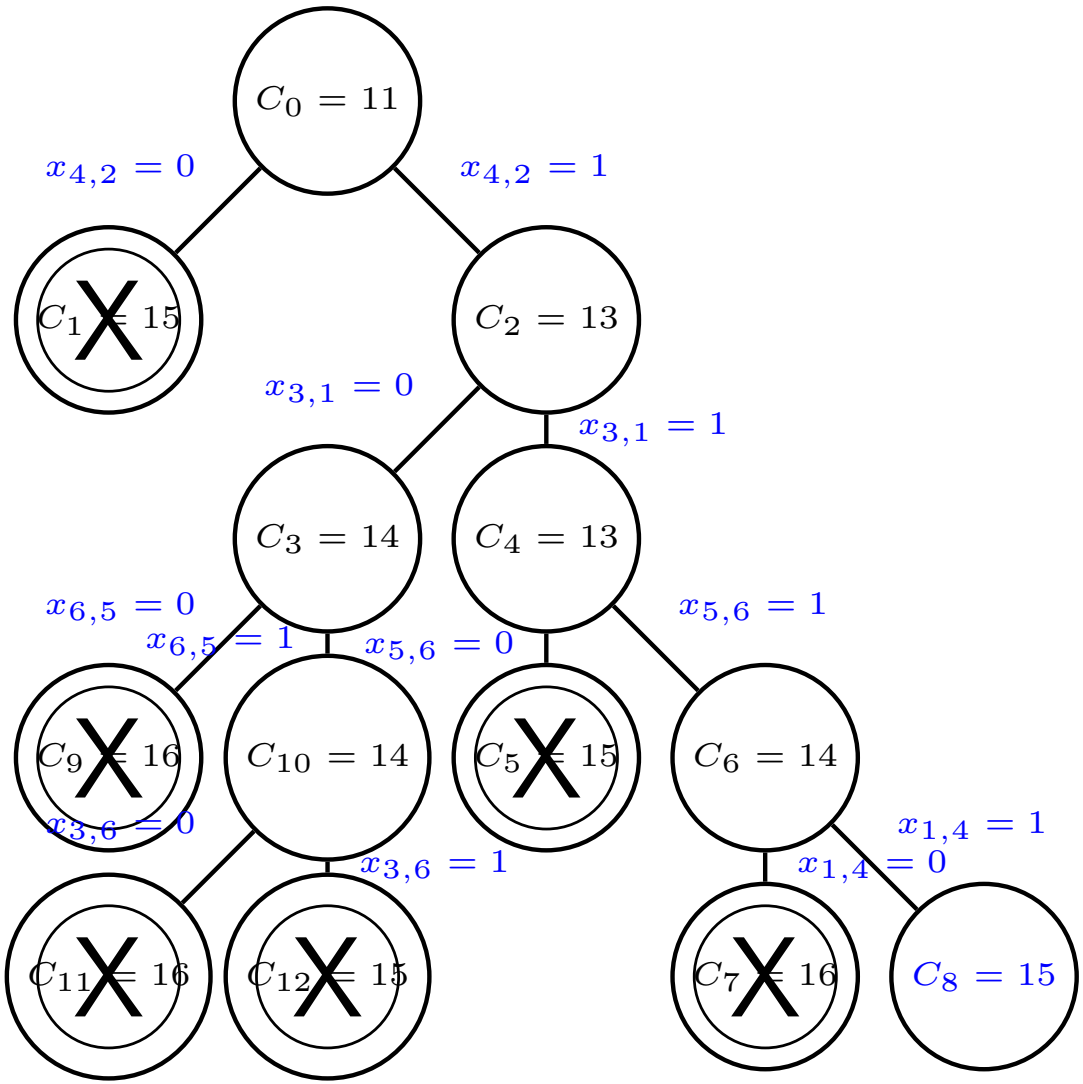
ATSP: Little et al. BB - Example

Compute the lower bound at node 12
((3,6)(6,5) $\rightarrow$ forbid (5,3))

	1	**3**	**4**	
1	M	3	0	
2	0	0	M	
5	1	M	1	-1

$C_{12} = C_{10} + 1 = 15$

ATSP: Little et al. BB - Example



prune nodes 11 and 12 $\rightarrow$ optimality proved

ATSP: Carpaneto-Toth BB algorithm

Notation

k . . .node in the BB tree

E_k . . .set of excluded arcs

I_k . . .set of included arcs

Q . . .queue of unprocessed nodes

Source: G. Carpaneto, P. Toth (1980) Some new branching and bounding criteria for the asymmetric travelling salesman problem. Management Science 26:736–743.

E. Balas, P. Toth (1990) Branch and bound methods. In E.L. Lawler J.K. Lenstra, A.H.G. Rinnooy Kan, D.B.

Shmoys (Eds) 'The traveling salesman problem'.

ATSP: Carpaneto-Toth BB algorithm

a slightly simplified version

1. **relaxation (lower bound)** modified assignment problem (MAP); AP with additional constraints: excluded and included arcs.
2. **choose a node** k from Q (the one with the smallest cost value of the associated MAP).
3. **branching scheme** at node k choose the subtour q with the minimum number of arcs not included in I_k. A_q is now the set of not included arcs of subtour q, $A_q = \{(i_1, f_1), (i_2, f_2), \ldots, (i_v, f_v)\}$. v child nodes are created. For each child node $r = 1, \ldots, v$ $(v + 1 = 1)$

$$E_{kr} = E_k \cup \{(i_r, i_{r+1})\}$$
$$I_{kr} = I_k \cup \{(i_1, i_2), \ldots, (i_{r-1}, i_r)\}$$

ATSP: Carpaneto-Toth BB - Example

Root node $k = 0$

$I_0 = \{\}$, $E_0 = \{\}$, $C_0^{MAP} = ?$ (lower bound)

	1	**2**	**3**	**4**	**5**	**6**	
1	M	7	4	2	1	3	-1
2	3	M	3	2	4	6	-2
3	2	3	M	4	5	3	-2
4	7	1	5	M	4	4	-1
5	4	4	3	5	M	3	-3
6	4	3	3	6	2	M	-2

ATSP: Carpaneto-Toth BB - Example

Root node $k = 0$

$I_0 = \{\}$, $E_0 = \{\}$, $C_0^{MAP} = ?$ (lower bound)

	1	**2**	**3**	**4**	**5**	**6**	
1	M	6	3	1	0	2	X
2	1	M	1	0	2	4	
3	0	1	M	2	3	1	
4	6	0	4	M	3	3	
5	1	1	0	2	M	~~0~~	
6	2	1	1	4	~~0~~	M	X
					X		

reduction constant = 1+2+2+1+3+2=11

ATSP: Carpaneto-Toth BB - Example

Root node $k = 0$

$I_0 = \{\}$, $E_0 = \{\}$, $C_0^{MAP} = ?$ (lower bound)

	1	**2**	**3**	**4**	**5**	**6**	
1	M	5	2	0	0	1	X
2	1	M	1	0	3	4	
3	0	1	M	2	4	1	
4	6	0	4	M	4	3	
5	1	1	0	2	M	0	
6	1	0	0	3	0	M	X
					X		

reduction constant = 1+2+2+1+3+2=11

smallest value in uncovered cells: 1

ATSP: Carpaneto-Toth BB - Example

Root node $k = 0$

$I_0 = \{\}$, $E_0 = \{\}$, $C_0^{MAP} = ?$ (lower bound)

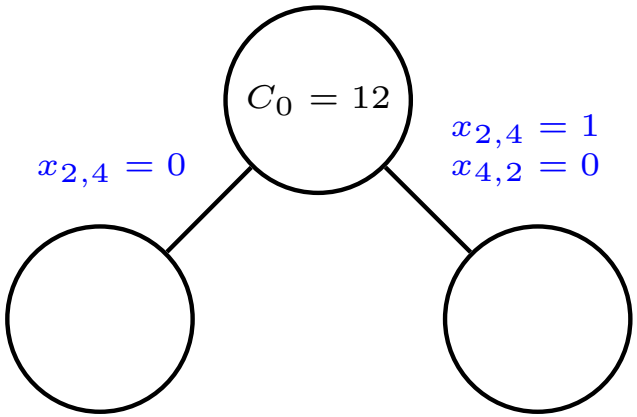
	1	**2**	**3**	**4**	**5**	**6**
1	M	5	2	~~0~~	0	1
2	1	M	1	0	3	4
3	0	1	M	2	4	1
4	6	0	4	M	4	3
5	1	1	~~0~~	2	M	0
6	1	~~0~~	0	3	~~0~~	M

reduction constant = 1+2+2+1+3+2+1=12

$C_0^{MAP} = 12$

Subcycles:
1-5-6-3-1
2-4-2
branch on 2-4-2

ATSP: Carpaneto-Toth BB - Example



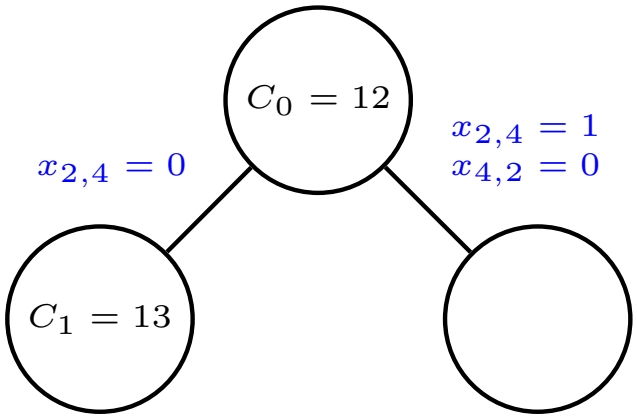
$k = 0$ **Node 1 is generated**
$I_1 = \{\}$, $E_1 = \{(2,4)\}$,
$C_1^{MAP} = ?$

	1	2	3	4	5	6
1	M	5	2	0	~~0~~	1
2	~~0~~ 0 1	M	0 01	M	2	3
3	0	1	M	2	4	1
4	6	0	4	M	4	3
5	1	1	~~0~~	2	M	0
6	1	~~0~~	~~0~~	3	0	M

$C_1^{MAP} = 13$
1-4-2-3-1
5-6-5

$LB = 12$
$UB = \infty$

ATSP: Carpaneto-Toth BB - Example



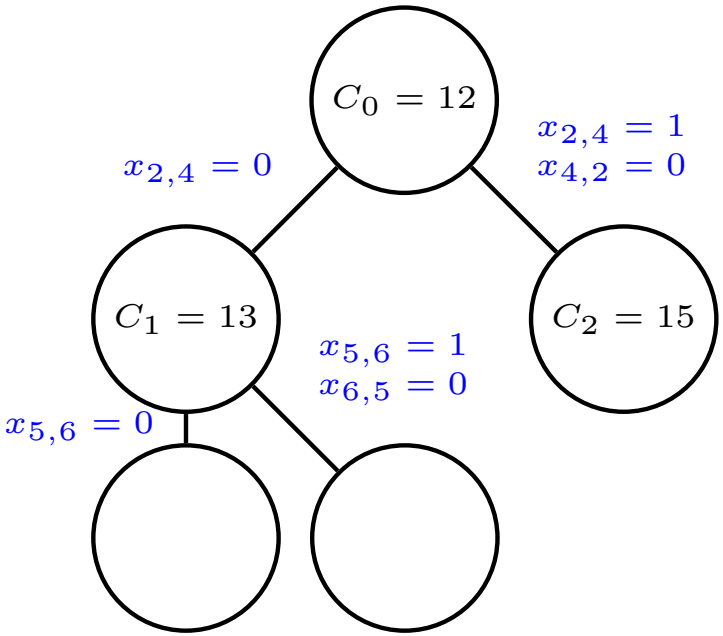
$LB = 13$
$UB = \infty$

$k = 0$ **Node 2 is generated**
$I_2 = \{(2,4)\}$, $E_2 = \{(4,2)\}$,
$C_2^{MAP} = ?$

	1	2	3	5	6	
1	M	5	2	0	1	
3	0	1	M	4	1	
4	2	M	3	3	0 03	-3
5	1	1	0	M	∞	
6	1	0	∞	∞	M	

$C_2^{MAP} = 15$
1-5-3-1
2-4-6-2

ATSP: Carpaneto-Toth BB - Example



$LB = 13$
$UB = \infty$

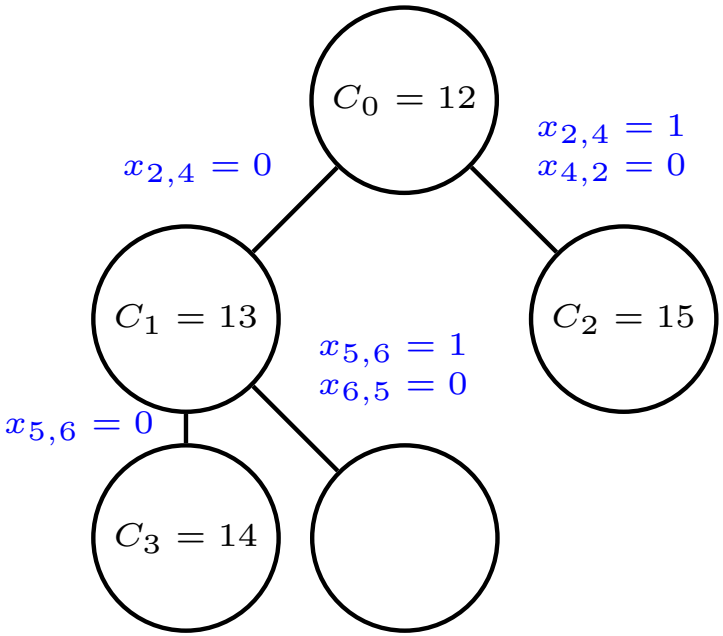
$k = 1$ **Node 3 is generated**
$I_3 = \{\}$, $E_3 = \{(2,4),(5,6)\}$,
$C_3^{MAP} = ?$

	1	2	3	4	5	6
1	M	5	2	0	~~0~~	~~0~~ 01
2	0	M	~~0~~	M	2	2
3	~~0~~	1	M	2	4	0 0 1
4	6	0	4	M	4	2
5	1	1	0	2	M	M
6	1	~~0~~	~~0~~	3	0	M

-1

$C_3^{MAP} = 14$
1-4-2-1
3-6-5-3

ATSP: Carpaneto-Toth BB - Example

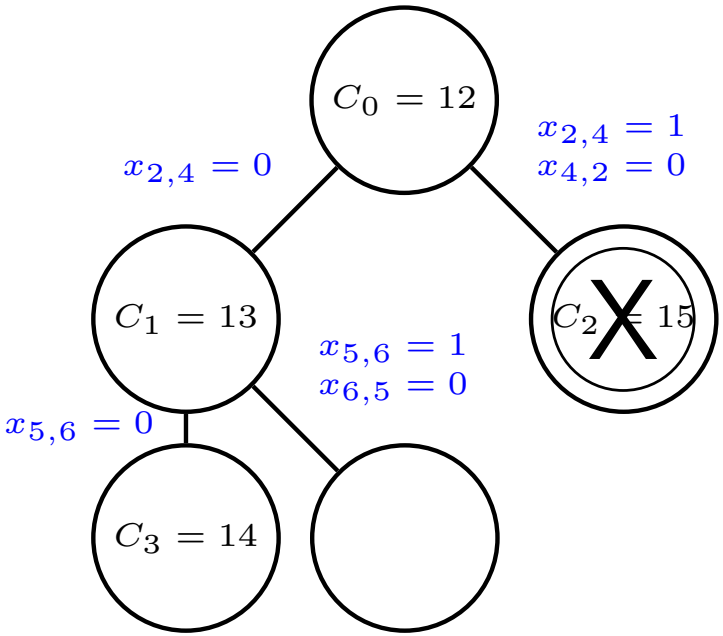


$k = 1$ **Node 4 is generated**
$I_4 = \{(5,6)\}$,
$E_4 = \{(2,4),(6,5)\}$, $C_4^{MAP} = ?$

	1	2	3	4	5	
1	M	5	2	0	~~0~~	
2	~~0~~	M	0	M	2	X
3	0	1	M	2	4	X
4	6	0	4	M	4	X
6	1	~~0~~	~~0~~	3	M	X
	X	X	X			

$LB = 13$
$UB = \infty$

ATSP: Carpaneto-Toth BB - Example



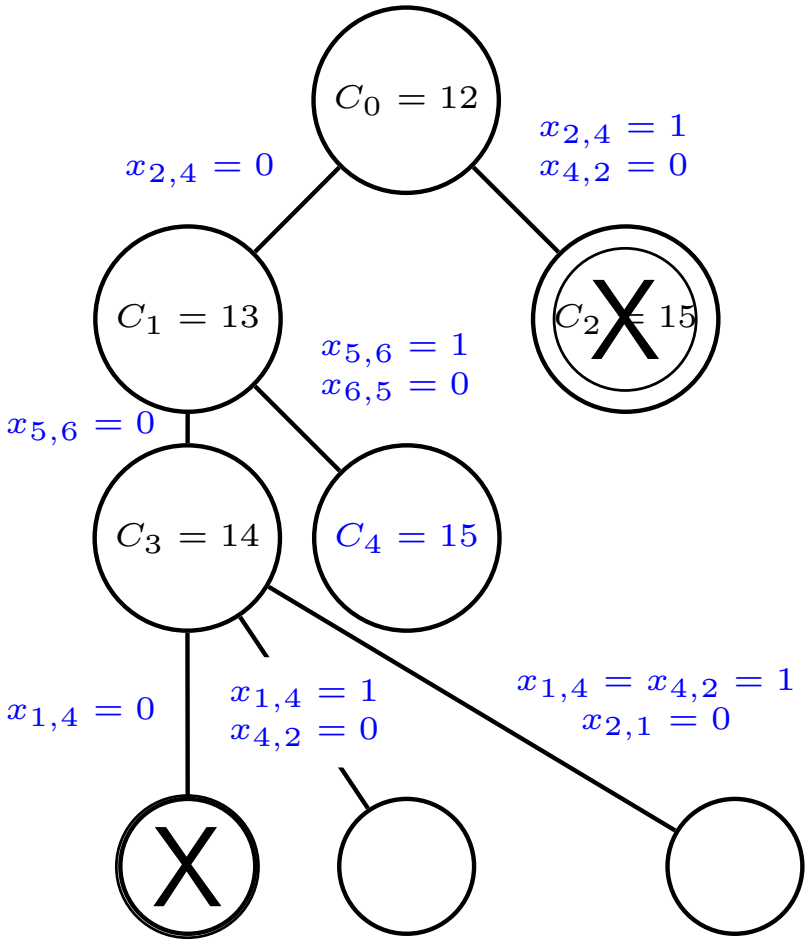
$LB = 14$
$UB = 15$

$k = 1$ **Node 4 is generated**
$I_4 = \{(5,6)\}$,
$E_4 = \{(2,4),(6,5)\}$, $C_4^{MAP} = ?$

	1	2	3	4	5	
1	M	7	4	~~0~~	0	
2	0	M	~~0~~	M	~~0~~02	X
3	~~0~~	1	M	0 02	2	X
4	6	0	4	M	2	X
6	1	~~0~~	0	1	M	X
	X	X	X			

smallest value in uncovered cells: 2
alternative solutions!
TSP: 1-5-6-3-4-2-1
$C_4^{TSP} = 15$

ATSP: Carpaneto-Toth BB - Example



$LB = 14$
$UB = 15$

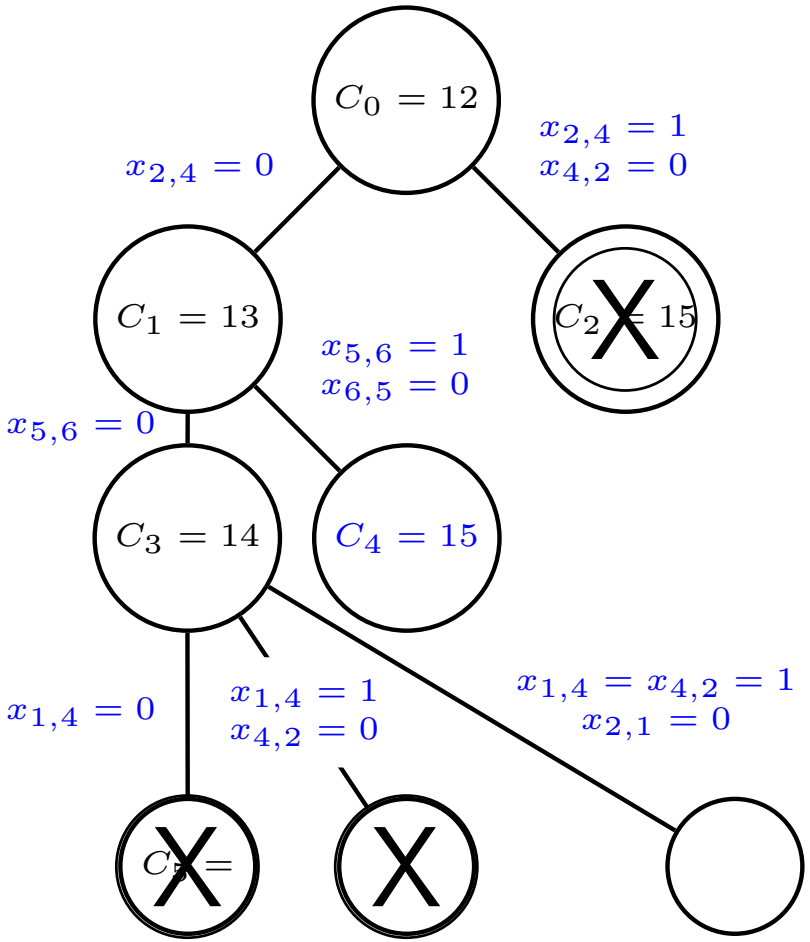
$k = 3$ **Node 5 is generated**
$I_5 = \{\}$,
$E_5 = \{(2,4),(5,6),(1,4)\}$,
$C_5^{MAP} = ?$

	1	2	3	4	5	6
1	M	5	2	M	0	0
2	0	M	0	M	2	2
3	0	1	M	2	4	0
4	6	0	4	M	4	2
5	1	1	0	2	M	M
6	1	0	0	3	0	M

-2

lower bound on $C_5^{MAP} = 16$
$16 > UB$**!**

ATSP: Carpaneto-Toth BB - Example

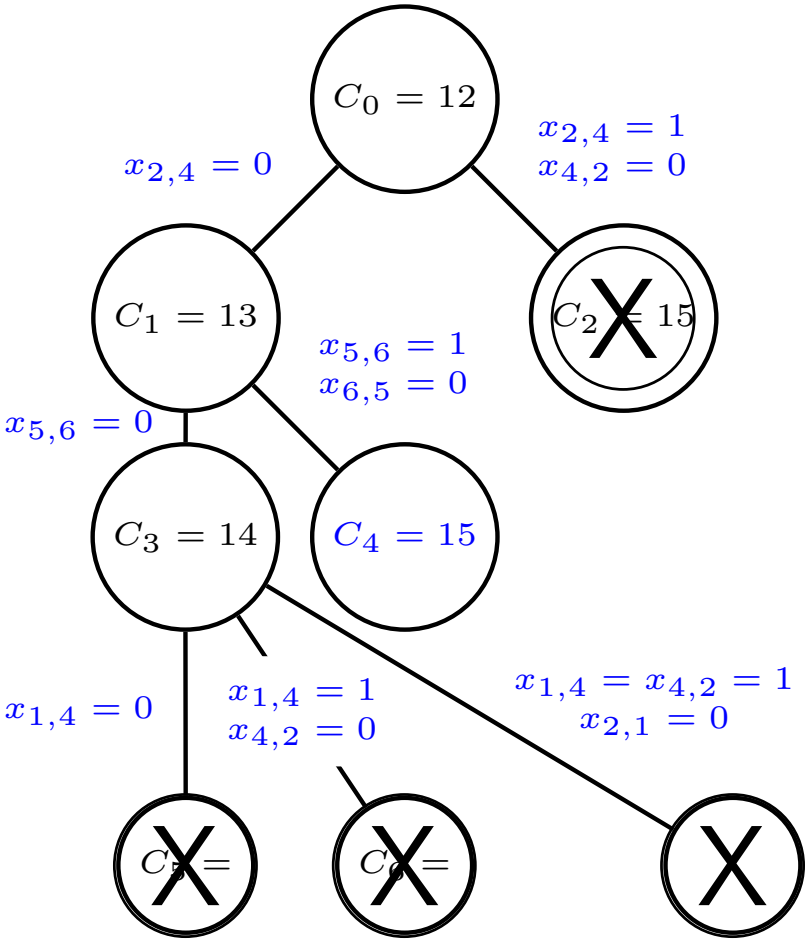


$LB = 14$
$UB = 15$

$k = 3$ **Node 6 is generated**
$I_6 = \{(1,4)\}$,
$E_6 = \{(2,4),(5,6)(4,2)\}$,
$C_6^{MAP} = ?$

	1	2	3	5	6	
2	0	M	0	2	2	
3	0	1	M	4	0	
4	6	M	4	4	2	-2
5	1	1	0	M	M	
6	1	0	0	0	M	

lower bound on $C_6^{MAP} = 16$
$16 > UB$**!**



$LB = 15$
$UB = 15$

$k = 3$ **Node 7 is generated**
$I_7 = \{(1,4),(4,2)\}$,
$E_7 = \{(2,4),(5,6),(2,1)\}$,
$C_7^{MAP} = ?$

	1	**3**	**5**	**6**
2	M	0	2	2
3	0	M	4	0
5	1	0	M	M
6	1	0	0	M

cost reduction: 1
lower bound on $C_7^{MAP} = 15$
$15 \geq UB$**!**

References

E.L. Lawler J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds) (1990) 'The traveling salesman problem', Wiley-Interscience Series in Discrete Mathematics and Optimization

G. Ghiani, G. Laporte, R. Musmano (2004) 'Introduction to Logistics Planning and Control' Wiley-Interscience Series in Systems and Optimization

W. Domschke (1997) 'Logistik: Rundreisen und Touren' Oldenbourg.